**Создание простой анимации блока.**

*Описание процесса создания простой графической визуализации блока.*

Анимация блоков довольно существенно упрощает работу с расчетной схемой, что особенно заметно при работе со сложными разветвленными схемами с большим количеством расчетных блоков. Например, при работе с расчетными схемами для теплогидравлических сетей очень часто необходимо знать в каком состоянии находится то или иное оборудование: включены или выключены насосы, закрыты или открыты задвижки; проконтролировать положение регулирующих клапанов, проверить текущие параметры схемы. Для схем алгоритмов важно знать в каком состоянии находятся функциональные блоки, например, такие как триггеры, а также значения передаваемые по линиям связи. Электрические схемы должны отображать состояние такого оборудования, как выключатели, двигатели и пр.

В случае небольших схем простой топологии это, конечно, можно осуществить и без визуализации блоков, просто просматривая результаты расчета блоков, тем самым создавая цельную картину расчета схемы. Однако даже в таком случае создание цельной картины расчета схемы возможно лишь при проведении стационарных режимов, где параметры слабо меняются с течением времени, или нестационарных режимов с медленным развитием процесса (развитие процесса занимает десятки минут и выше). При динамичном же нестационарном режиме, когда развитие какого-либо процесса занимает секунды, за расчетными параметрами блоков уследить практически невозможно, что приводит к невозможности анализа и управления пользователем данного процесса.

В случае же крупных схем с разветвленной топологией и большим количеством моделируемого оборудования провести наблюдения за ходом расчета без визуализации блоков невозможно.

Прежде чем визуализировать расчетный блок, необходимо определиться с ответами на следующие вопросы:

* какие расчетные параметры должны быть визуализированы?
* каким образом будет осуществлена визуализация: с помощью отображения расчетных параметров блока (например, для реализации непрерывного отображения параметров расчетных узлов, показаний датчиков и пр.), с помощью цветовой индикации, с помощью изменения графического изображения блока, и т.п.?
* будет ли понятной конечному пользователю блока вносимая визуализация?

После ответов на данные вопросы можно приступать к визуализации блока.

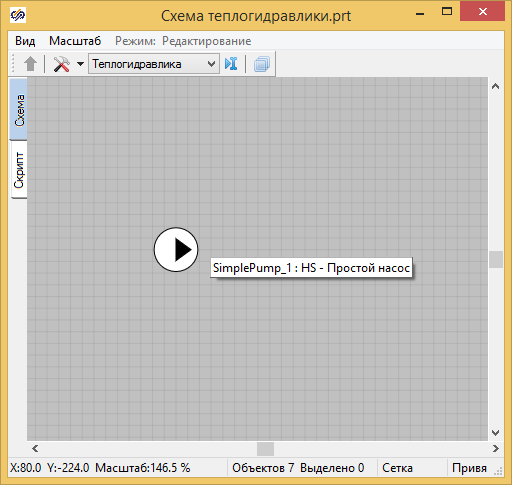
Процесс визуализации блока предусматривает следующие этапы редактирования блока:

1. Внесение дополнительных свойств, параметров, локальных переменных для блоков – при необходимости;
2. Внесение дополнительных примитивов – так же при необходимости;
3. Написание скриптов визуализации блока;
4. Тестирование блока на правильность работы визуализации;
5. Занесение блока в библиотеку – в случае, если блоком пользуется не единичный пользователь.

Прохождение данных этапов позволяет получить законченный блок с простой анимацией.

В качестве примера рассмотрим процесс визуализации блока «HS – Простой насос», принадлежащего библиотеке HS.

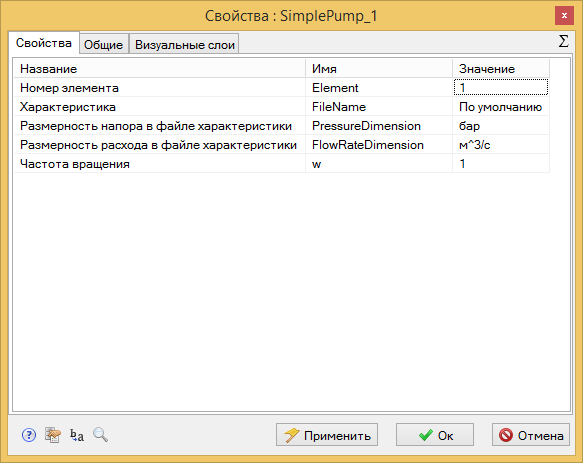
Исходный блок, находящийся в библиотеке, имеет такой вид (см. **Рисунок 1**):



**Рисунок 1. Внешний вид блока «HS – Простой насос»**

Исходя из первичного визуального анализа данного блока становится понятно, что для пользователя необходима визуализация статуса насоса, т.е. включен или отключен насос.

Теперь зайдем в свойства данного блока (см. **Рисунок 2**).



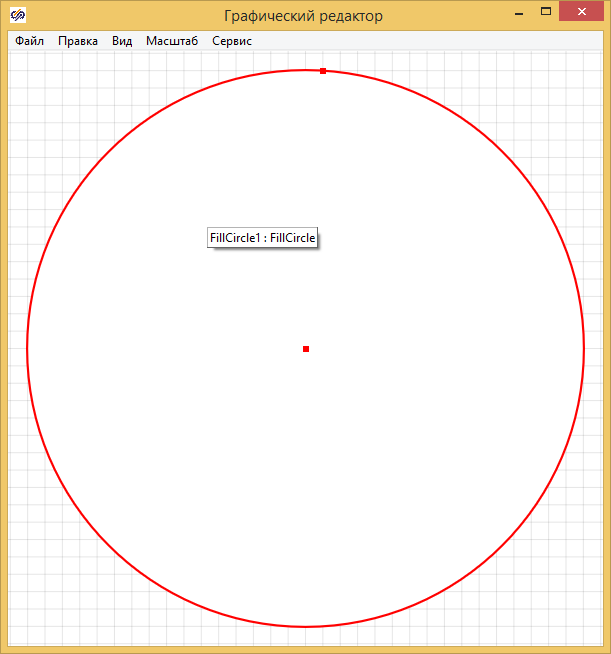
**Рисунок 2. Свойства блока «HS – Простой насос»**

Из приведенной таблицы свойств видно, что свойство c именем «**w**» (и названием «**Частота вращения**») как раз и может помочь нам в визуализации данного блока: данное свойство отвечает за относительную частоту вращения насоса и изменяется от 0 до 1. Упрощенно примем, что при частоте вращения выше определенного уровня можно считать, что насос включен, а ниже того же уровня, что насос отключен.

Теперь зайдем в редактор графической оболочки блока и рассмотрим примитивы, с помощью которых был сделан данный блок. Для этого необходимо перейти на закладку «**Общие**» → «**Графическое изображение**».

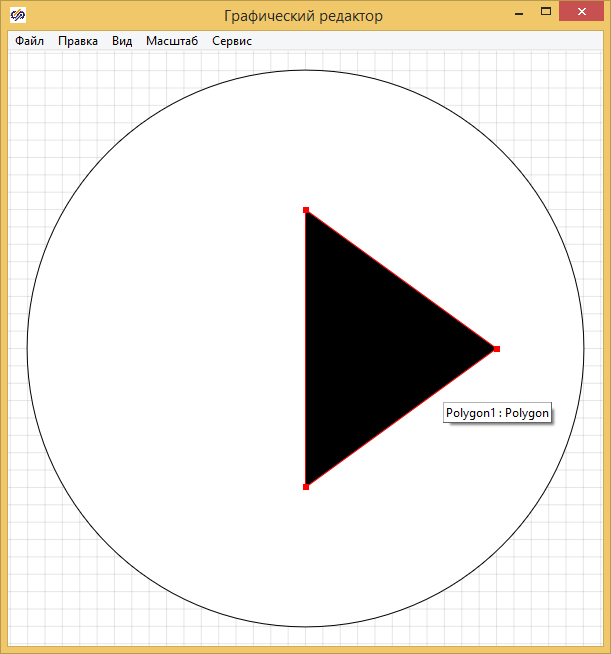
Из нижеприведённых рисунков следует, что графическое изображение блока состоит из двух примитивов:

* примитива «**Круг**» (см. **Рисунок 3**)



**Рисунок 3. Состав графического изображения блока – примитив «Круг»**

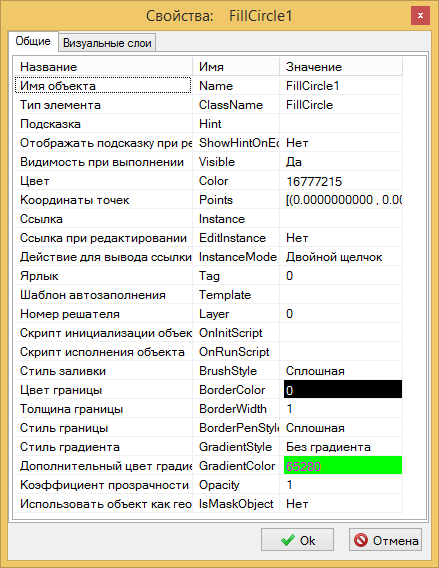
* примитива «**Полигон**» (см. **Рисунок 4**)



**Рисунок 4. Состав графического изображения блока – примитив «Полигон»**

Если выбирать между двумя примитивами, то наиболее заметным для пользователя при визуальном изменении будет примитив «**Круг**», значит с ним и желательно работать.

Заходим в свойства примитива и решаем с какими будем работать (см. **Рисунок 5**).



**Рисунок 5. Свойства графического примитива «Круг»**

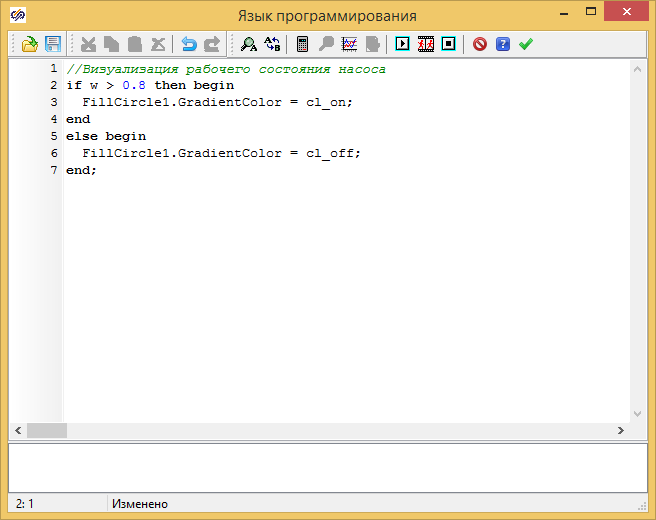
Довольно очевидно, что наилучшим способом привлечь внимание пользователя при изменении статуса объекта является изменение цветового содержания примитива.

За основной цвет примитива отвечает свойство **Color**, однако мы сделаем визуализацию нашего блока чуть более интересной и добавим градиент к основному цвету. За дополнительный цвет градиента отвечает свойство **GradientColor**.

Общепринятой индикацией включенного насоса является красный цвет, а выключенного насоса – зеленый.

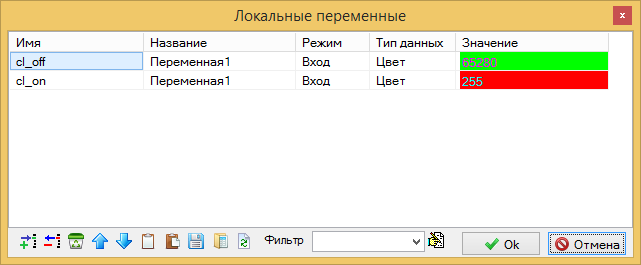
Исходя из вышесказанного нам нужно составить такой скрипт, чтобы он влиял на изменение цвета примитива таким образом, чтобы во включенном состоянии он был красным, а в отключенном – зеленым.

Пишем скрипт визуализации (см. **Рисунок 6**). Для того, чтобы зайти в редактор скрипта необходимо, находясь в графическом редакторе, пройти по пути «**Сервис**» → «**Скрипт…**»



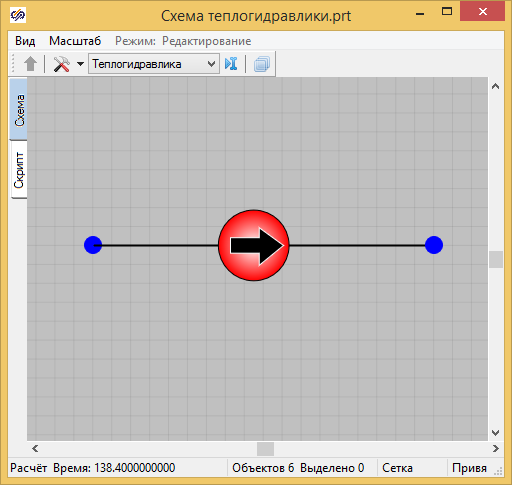
**Рисунок 6. Скрипт визуализации**

Для того, чтобы данный скрипт правильно необходимо еще добавить две локальных переменных в «**Сервис**» → «**Локальные переменные…**» (см. **Рисунок 7**), отвечающие за изменение цвета.



**Рисунок 7. Локальные переменные для работы скрипта**

При необходимости добавляем либо изменяем в графическом изображении примитивы и сохраняем внесенные изменения. В итоге получается блок, изменяющий свой вид в соответствии со значением внутреннего свойства. На **рисунке 8** показан блок «HS – Простой насос» с относительной частотой вращения **w**, равной 1 (т.е. во включенном состоянии).



**Рисунок 8. Блок «HS – Простой насос» с активированной визуализацией**